



INSTALLATION RUNBOOK FOR MOS + Hillstone FWaaS Plugin Driver

Application Type: **Firewall Policy Management**

Application Version: **1.0**

MOS Version: **7.0**

OpenStack version: **Kilo**

Content

[Document History](#)

[1 Introduction](#)

[1.1 Target Audience](#)

[2 Application overview](#)

[3 Joint Reference Architecture](#)

[4 Physical & Logical Network Topology](#)

[5 Installation & Configuration](#)

[5.1 Environment preparation](#)

[5.2 MOS installation](#)

[5.2.1 Health Check Results](#)

[5.3 Hillstone FWaaS Plugin Driver installation steps](#)

[5.4 Limitations](#)

[5.5 Testing](#)

[5.5.1 Test cases](#)

[5.5.2 Test Results](#)

Document History

Version	Revision Date	Description
1.0	27-04-2016	Initial Version

1 Introduction

The Hillstone FWaaS Plugin Driver enables policy configuration of Hillstone firewall appliances through OpenStack Neutron FWaaS. This document provides a validated reference architecture and offers step-by-step installation instructions for integrating the Hillstone FWaaS Plugin Driver with Mirantis OpenStack. We also discuss limitations of the integration, and describe testing procedures used in validation.

1.1 Target Audience

This document is for OpenStack data center admins, or data center security admins.

2 Application overview

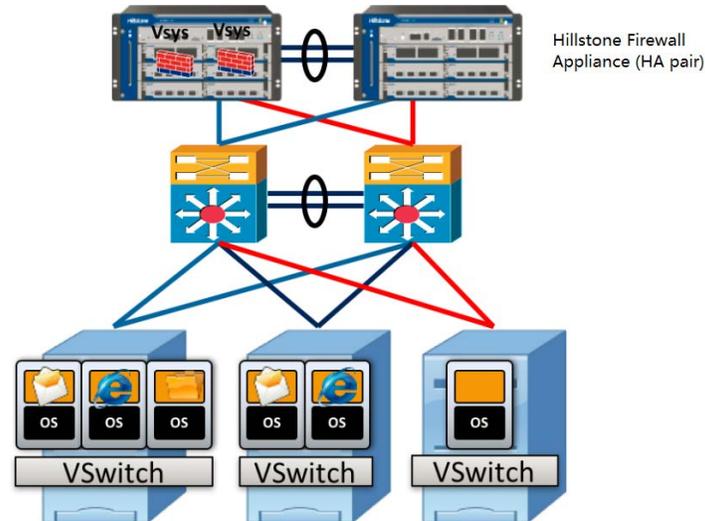
The Hillstone FWaaS (Firewall as a Service) Plugin Driver takes firewall policy configuration from the OpenStack Neutron FWaaS extension, converts it to Hillstone policy format, and sends it to a Hillstone firewall appliance deployed in an OpenStack data center.

Hillstone Networks provides a high performance firewall appliance that can be deployed at the perimeter of a data center to secure North-South traffic. The Firewall appliance provides Web UI, CLI and RESTful API management interfaces for admins to configure firewall policies. With the Hillstone FWaaS Plugin Driver deployed in Mirantis OpenStack, the OpenStack admin can configure firewall policies through the FWaaS plugin, and policies will be pushed to the Hillstone firewall appliance automatically.

OpenStack security groups and the FWaaS plugin can only provide firewall protection on Layer 3 to Layer 4. As a full-featured firewall, the Hillstone firewall appliance can provide not only Layer 2 to L4 firewall, but also offers several Layer 7 protections like Application Identification, IPS, AV, URL filtering, etc. The Firewall appliance can be deployed at the data center perimeter to secure North-South traffic.

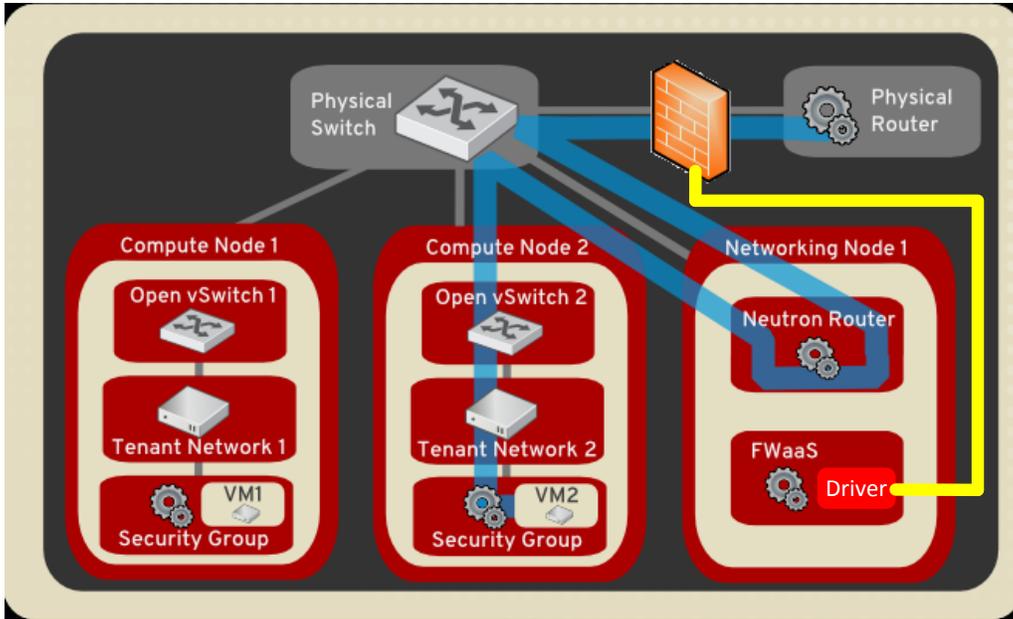
3 Joint Reference Architecture

The Hillstone firewall appliance can be deployed at the perimeter of data center as shown in the following figure:



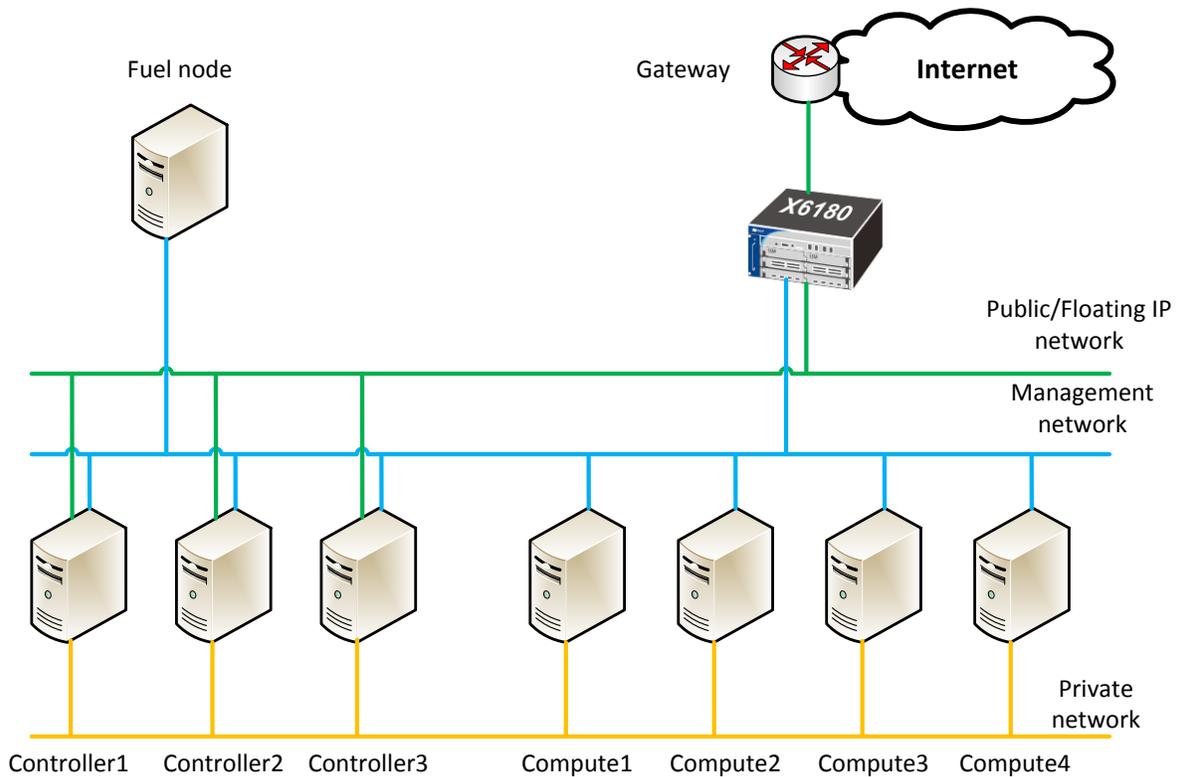
More detail about the Hillstone data center firewall appliance can be found at <http://www.hillstonenet.com/our-products/datacenter-next-gen-firewalls-x-series/>.

When integrated with Mirantis OpenStack, the firewall appliance is deployed at the data center perimeter, i.e., between OpenStack's external network and the gateway/router to the Internet. The Hillstone FWaaS Plugin Driver is deployed on all network/neutron nodes, and the OpenStack Neutron FWaaS extension is configured to point to it. FWaaS firewall rule configuration sent by the FWaaS Neutron extension to L3-agent is converted to a Hillstone configuration API request and sent to the firewall appliance (shown as the yellow path in the following architecture diagram). In the current release, only one tenant can configure the FWaaS firewall, and firewall policy configuration is based on IPs on the public/floating network.

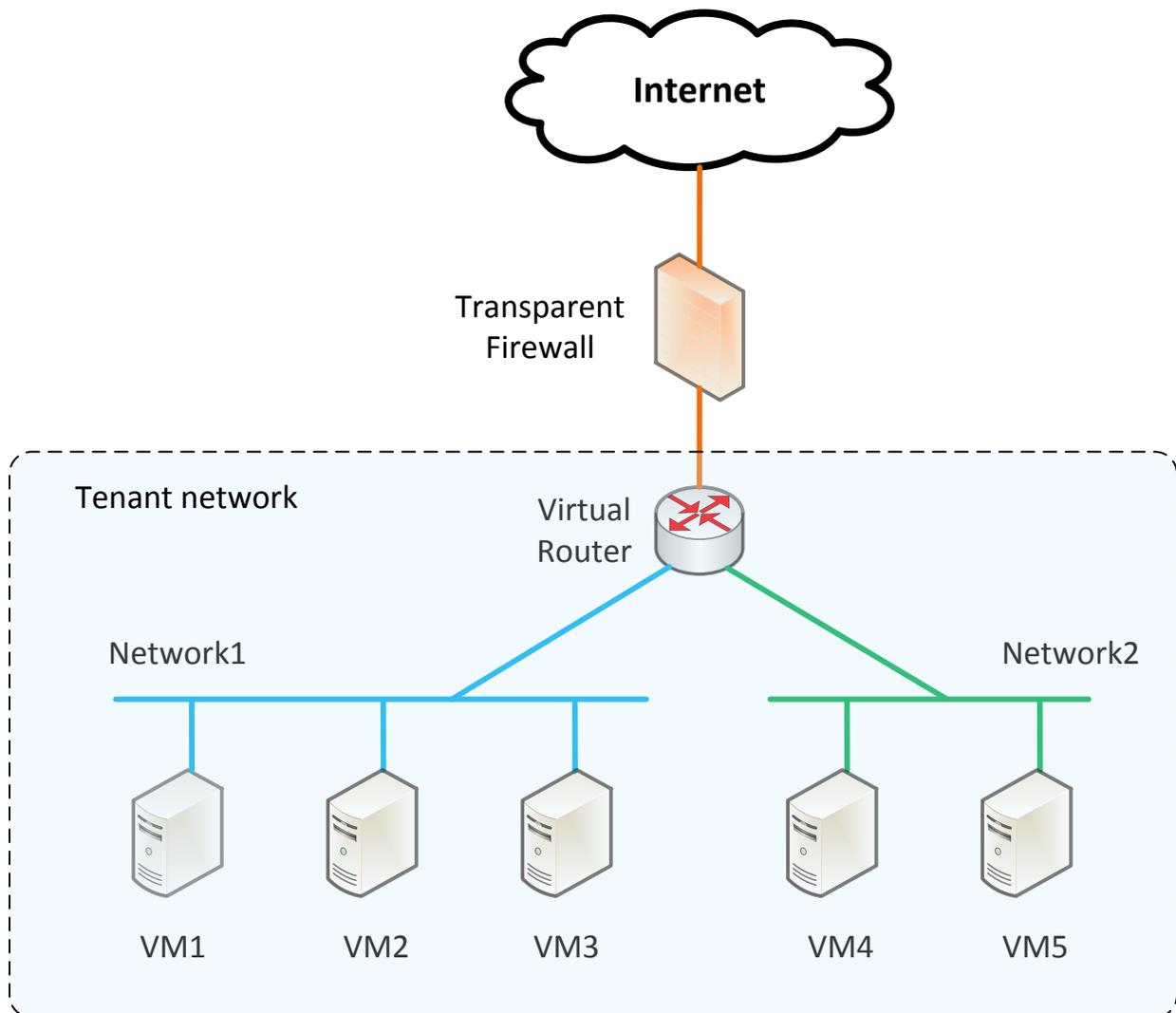


4 Physical & Logical Network Topology

Physical network topology is shown in the following diagram. Compared with a standard Mirantis OpenStack deployment, a Hillstone firewall appliance is deployed between the Public network and the Gateway as a perimeter firewall. The firewall management interface connects to the data center Management network.



The logical network topology is shown in the following diagram. Each tenant manages his own networks, VMs, and virtual routers. The Hillstone firewall is deployed in transparent mode between the tenant virtual router and the Internet. Firewall rules configured through OpenStack's FWaaS plugin are automatically synced to the Hillstone firewall.



5 Installation & Configuration

5.1 Environment preparation

The Hillstone FWaaS Plugin Driver is installed to an existing Mirantis OpenStack 7.0 data center that has the OpenStack Neutron FWaaS extension enabled, and a Hillstone firewall appliance installed. The following instructions provide a (high level) overview of how such a data center

may be deployed. Rack space and power must naturally be provided for the Hillstone firewall appliance (please refer to Hillstone's data sheet for details).

5.2 MOS installation

Download and install the Mirantis OpenStack 7.0 .iso and create a Fuel Master (OpenStack cluster deployer) by following Mirantis' instructions:

<https://docs.mirantis.com/openstack/fuel/fuel-7.0/>

Also enable the Fuel Master to deploy the OpenStack FWaaS Neutron extension by installing the FWaaS Fuel Plugin, as detailed in the following doc:

<http://plugins.mirantis.com/docs/f/w/fwaas-plugin/fwaas-plugin-1.1-1.1.0-1.pdf>

After the FWaaS Fuel Plugin is installed to the Fuel Master, you can use Fuel to configure and create a new OpenStack environment. The environment tested in this document was configured as follows:

Mirantis Openstack version: MOS7.0 Kilo on Ubuntu 14.04

Networking backend: Neutron with VLANs

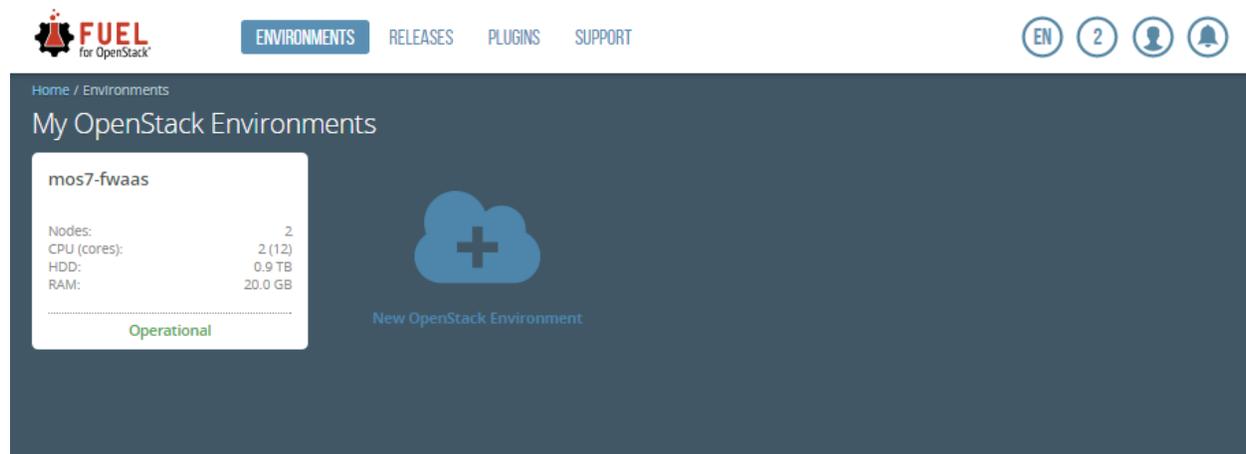
Glance backend: Swift

Cinder backend: LVM but no Cinder nodes were installed.

Nodes in the environment: 1 Controller node and 1 Compute node.

FWaaS plugin version: 1.1-1.1.0

In this sample setup, the OpenStack environment is called mos7-fwaas.



The following screen shots show more detail about this OpenStack environment.

mos7-fwaas (2 nodes)

- [Dashboard](#)
- [Nodes](#)
- [Networks](#)
- [Settings](#)
- [Logs](#)
- [Health Check](#)

Success

Deployment of environment 'mos7-fwaas' is done.

Plugin fwaas-plugin is deployed. Neutron extension that introduces Firewall feature set

Horizon

OpenStack Environment management panel (Horizon) is now available

[Proceed to Horizon](#)

Summary

Name	mos7-fwaas
Status	Operational
Openstack Release	Kilo on Ubuntu 14.04
Compute	KVM
Network	Neutron with VLAN segmentation
Storage Backends	Cinder LVM over iSCSI for volumes

To check out the OpenStack Healthcheck status go to [Healthcheck tab](#)

[Delete Environment](#) [Reset Environment](#)

Capacity

CPU (Cores)	12	HDD	0.9 TB	RAM	20.0 GB
-------------	----	-----	--------	-----	---------

Node Statistics

Total Nodes	2	Ready	2
Controller	1		
Compute	1		
Storage - Cinder	1		
Operating System	2		

[+ Add nodes](#)

Documentation

Quick access to the documentation on configuring and deploying environment:

- [Mirantis Openstack Documentation](#)
- [Plugin Documentation](#)
- [Technical Bulletins](#)

mos7-fwaas (2 nodes)

[Dashboard](#) [Nodes](#) [Networks](#) [Settings](#) [Logs](#) [Health Check](#)

[Configure Disks](#)[Configure Interfaces](#)[+ Add Nodes](#)

Sort By Roles ↓

 Select All

Controller, Operating System (1)

 Select All

<input type="checkbox"/>	mos7-controller CONTROLLER · BASE-OS		READY	CPU: 1 (4) HDD: 0.5 TB RAM: 4.0 GB
--------------------------	--	--	-------	------------------------------------

Compute, Storage - Cinder, Operating System (1)

 Select All

<input type="checkbox"/>	mos7-compute1 COMPUTE · CINDER · BASE-OS		READY	CPU: 1 (8) HDD: 0.5 TB RAM: 16.0 GB
--------------------------	--	--	-------	-------------------------------------



Network Settings

Neutron with VLAN segmentation

Public

IP Range	Start 10.1.1.100	End 10.1.1.109
CIDR	10.1.1.0/24	
Use VLAN tagging	<input type="checkbox"/>	
Gateway	10.1.1.253	
Floating IP ranges	Start 10.1.1.110	End 10.1.1.119

Storage

CIDR	192.168.21.0/24
Use VLAN tagging	<input checked="" type="checkbox"/> 21

Management

CIDR	192.168.20.0/24
Use VLAN tagging	<input checked="" type="checkbox"/> 20

Neutron L2 Configuration

VLAN ID range	110	119
Base MAC address	fa:16:3e:00:00:00	

Neutron L3 Configuration

Internal network CIDR	192.168.111.0/24	
Internal network gateway	192.168.111.1	
Guest OS DNS Servers	8.8.4.4	<input type="checkbox"/>
	8.8.8.8	<input type="checkbox"/>



Verification succeeded. Your network is configured correctly.

Network verification performs the following checks:

1. L2 connectivity checks between every node in the environment.
2. DHCP discover check on all nodes.
3. Packages repo connectivity check from master node.
4. Packages repo connectivity check from slave nodes via public & admin (PXE) networks.

[Verify Networks](#) [Cancel Changes](#) [Save Settings](#)

Home / Environments / mos7-fwaas / Settings

mos7-fwaas (2 nodes)

Dashboard Nodes Networks Settings Logs Health Check

OpenStack Settings

Access FWaaS plugin for Neutron

Additional Components

Common

Kernel parameters

Neutron Advanced Configuration

Repositories

Syslog

Public network assignment

Storage

FWaaS plugin for Neutron

Host OS DNS Servers

Host OS NTP Servers

Public TLS

Load Defaults Cancel Changes Save Settings

After installing the FWaaS Plugin on the Fuel Master node, you must enable it by checking “FWaaS plugin for Neutron” on Fuel's Settings page before deploying the environment. Other options on the Settings page can remain as default.

After deploying this environment, login to mos7-fwaas Horizon, and double check that the Firewalls tab is available under Network.

openstack admin

Overview

Limit Summary

Instances Used 0 of No Limit	VCPUs Used 0 of No Limit	RAM Used 0Bytes of No Limit
Volumes Used 1 of 10	Volume Storage Used 1GB of 1000GB	

Project ^
Compute v
Network ^
Network Topology
Networks
Routers
Firewalls
Object Store v
Orchestration v
Admin v
Identity v

5.2.1 Health Check Results

The Health Check results are shown in the following:

OpenStack Health Check

 Select All

<input type="checkbox"/>		Expected Duration	Actual Duration	Status
Sanity tests. Duration 30 sec - 2 min				
<input type="checkbox"/>	Request flavor list	20 s.	8.8	✓
<input type="checkbox"/>	Request image list using Nova	20 s.	0.7	✓
<input type="checkbox"/>	Request instance list	20 s.	2.2	✓
<input type="checkbox"/>	Request absolute limits list	20 s.	2.1	✓
<input type="checkbox"/>	Request snapshot list	20 s.	16.6	✓
<input type="checkbox"/>	Request volume list	20 s.	8.1	✓
<input type="checkbox"/>	Request image list using Glance v1	10 s.	0.1	✓
<input type="checkbox"/>	Request image list using Glance v2	10 s.	0.0	✓
<input type="checkbox"/>	Request stack list	20 s.	0.0	✓
<input type="checkbox"/>	Request active services list	20 s.	3.2	✓
<input type="checkbox"/>	Request user list	20 s.	2.9	✓
<input type="checkbox"/>	Check that required services are running	180 s.	12.7	✓
<input type="checkbox"/>	Request list of networks	20 s.	0.2	✓
Functional tests. Duration 3 min - 14 min				
<input type="checkbox"/>	Create instance flavor	30 s.	21.0	✓

5.3 Hillstone FWaaS Plugin Driver installation steps

Download the Hillstone FWaaS Plugin Driver, `hillstone-fwaas-driver-kilo-v1.0.tar.gz`, from the following url:

<http://www.hillstonenet.com/wp-content/uploads/hillstone-fwaas-driver-kilo-v1.0.tar.gz>

Save the tar file on the target cluster's main OpenStack Controller node. Extract files as follows:

```
root@node-2: ~  
root@node-2:~# tar xvf hillstone-fwaas-driver-kilo-v1.0.tar.gz  
hillstone-fwaas-driver-kilo-v1.0/  
hillstone-fwaas-driver-kilo-v1.0/.config1.swp  
hillstone-fwaas-driver-kilo-v1.0/README  
hillstone-fwaas-driver-kilo-v1.0/srcbk/  
hillstone-fwaas-driver-kilo-v1.0/hillstone_common/  
hillstone-fwaas-driver-kilo-v1.0/hillstone_common/README_For_Common  
hillstone-fwaas-driver-kilo-v1.0/hillstone_common/common/  
hillstone-fwaas-driver-kilo-v1.0/hillstone_common/common/hillstone_vfw_api.py  
hillstone-fwaas-driver-kilo-v1.0/hillstone_common/common/__init__.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/hillstone_driver_exception.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/util.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/odl/  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/odl/odl_resource.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/odl/neutron_resource.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/odl/__init__.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/hillstone_manager.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/hillstone_fwaas.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/__init__.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/hillstone/hillstone_cfg.py  
hillstone-fwaas-driver-kilo-v1.0/fwaas/config/  
hillstone-fwaas-driver-kilo-v1.0/fwaas/config/fwaas_driver_tmpl.ini  
hillstone-fwaas-driver-kilo-v1.0/install-hs-fwaas-driver.sh  
root@node-2:~#
```

Run the following command as root to install the Hillstone FWaaS Plugin Driver:

```
root@node-2: ~/hillstone-fwaas-driver-kilo-v1.0  
root@node-2:~# cd hillstone-fwaas-driver-kilo-v1.0/  
root@node-2:~/hillstone-fwaas-driver-kilo-v1.0# ./install-hs-fwaas-driver.sh install
```

During installation, the installation script asks the admin to provide several inputs. Details of each input are as follows:

Firewall Management IP

Management IP for the Hillstone firewall appliance. This should be reachable from the Controller through the Management network.

Firewall Management http port

http port for the firewall appliance web management interface (default 80).

Firewall Management ssh port

ssh port for the firewall appliance cli management interface (default 22).

API method

API method the driver will use to communicate with the firewall appliance (default is RestAPI).

Hillstone Firewall deployment method

The driver can support multiple deployment methods for the Hillstone firewall. The deployment tested here was 'Firewall Appliance.'

OS_TENANT_NAME

OS_USERNAME

OS_PASSWORD

OS_AUTH_URL

OS_REGION_NAME

The above parameters authenticate the OpenStack admin user and toplevel tenant. They can be retrieved from the openrc.sh file downloaded from Horizon (Compute->Access & Security->API Access tab).

Protected tenant name

The tenant name that will be protected by the firewall.

Firewall port that connects to datacenter network

The name of the firewall port that connects to the data center network.

Firewall port that connects to Internet

The name of the firewall port that connects to the Internet.

Firewall admin user name

Firewall admin user name (default *hillstone*).

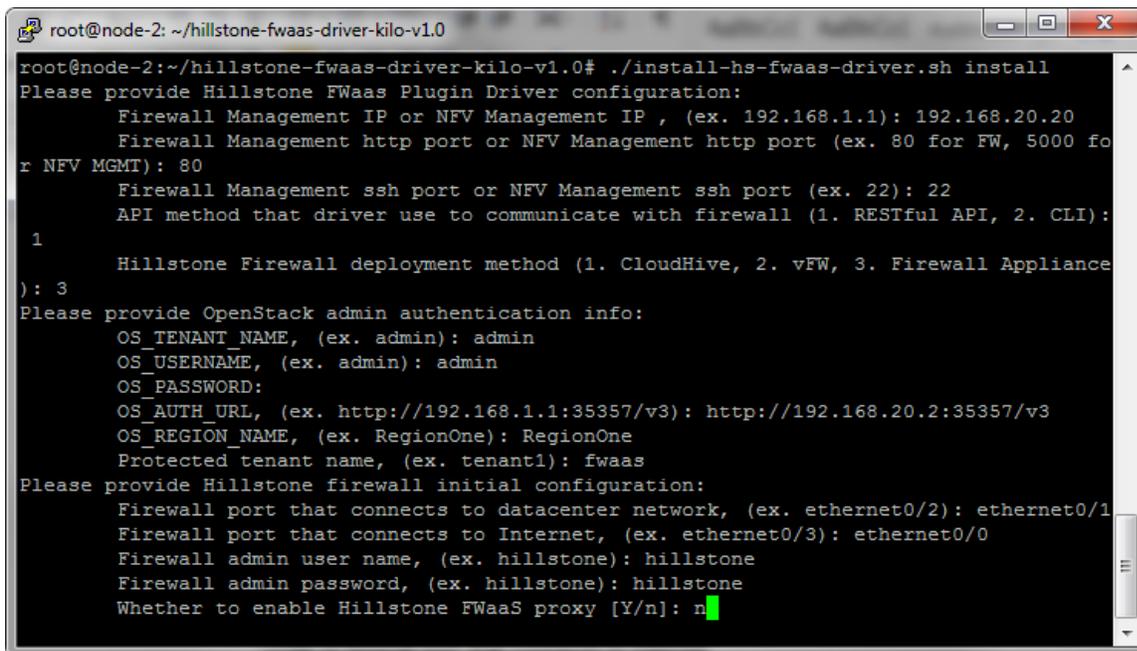
Firewall admin password

Firewall admin user password (default *hillstone*).

Whether to enable Hillstone FWaaS proxy

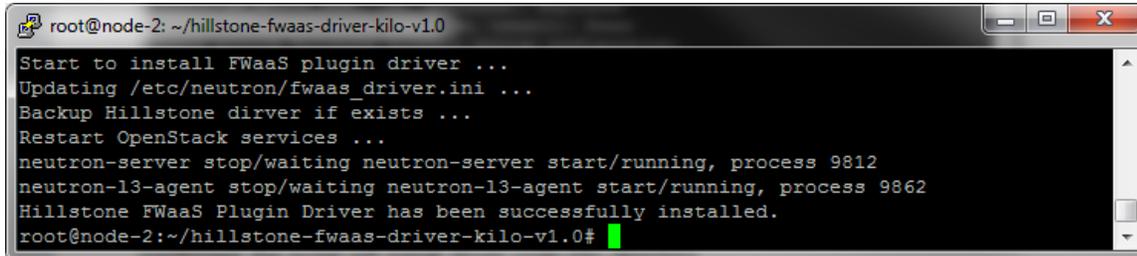
FWaaS proxy is not required in this deployment.

For this deployment, inputs were as follows:



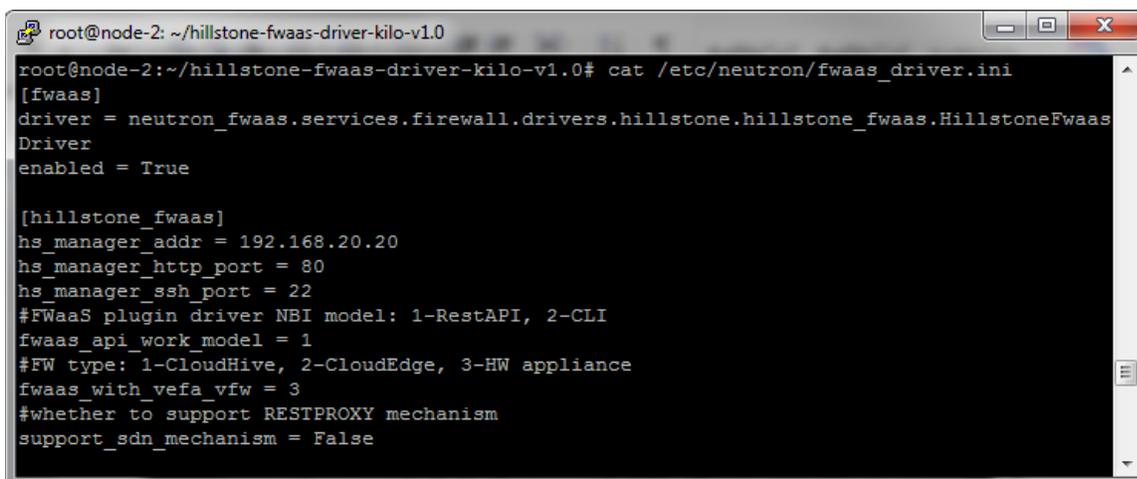
```
root@node-2: ~/hillstone-fwaas-driver-kilo-v1.0
root@node-2:~/hillstone-fwaas-driver-kilo-v1.0# ./install-hs-fwaas-driver.sh install
Please provide Hillstone FWaaS Plugin Driver configuration:
  Firewall Management IP or NFV Management IP , (ex. 192.168.1.1): 192.168.20.20
  Firewall Management http port or NFV Management http port (ex. 80 for FW, 5000 for NFV MGMT): 80
  Firewall Management ssh port or NFV Management ssh port (ex. 22): 22
  API method that driver use to communicate with firewall (1. RESTful API, 2. CLI):
  1
  Hillstone Firewall deployment method (1. CloudHive, 2. vFW, 3. Firewall Appliance): 3
Please provide OpenStack admin authentication info:
  OS_TENANT_NAME, (ex. admin): admin
  OS_USERNAME, (ex. admin): admin
  OS_PASSWORD:
  OS_AUTH URL, (ex. http://192.168.1.1:35357/v3): http://192.168.20.2:35357/v3
  OS_REGION_NAME, (ex. RegionOne): RegionOne
  Protected tenant name, (ex. tenant1): fwaas
Please provide Hillstone firewall initial configuration:
  Firewall port that connects to datacenter network, (ex. ethernet0/2): ethernet0/1
  Firewall port that connects to Internet, (ex. ethernet0/3): ethernet0/0
  Firewall admin user name, (ex. hillstone): hillstone
  Firewall admin password, (ex. hillstone): hillstone
  Whether to enable Hillstone FWaaS proxy [Y/n]: n
```

After input of the above parameters, the installation script will ask the admin to confirm the input. If confirmed, the script will install driver code into the directory `/usr/lib/python2.7/dist-packages/neutron_fwaas/services/firewall/drivers/`. Neutron-server and neutron-l3-agent services will be restarted at the end of installation.



```
root@node-2: ~/hillstone-fwaas-driver-kilo-v1.0
Start to install FWaaS plugin driver ...
Updating /etc/neutron/fwaas_driver.ini ...
Backup Hillstone dirver if exists ...
Restart OpenStack services ...
neutron-server stop/waiting neutron-server start/running, process 9812
neutron-l3-agent stop/waiting neutron-l3-agent start/running, process 9862
Hillstone FWaaS Plugin Driver has been successfully installed.
root@node-2:~/hillstone-fwaas-driver-kilo-v1.0#
```

To confirm the installation, admin can check `/etc/neutron/fwaas_driver.ini`.



```
root@node-2:~/hillstone-fwaas-driver-kilo-v1.0# cat /etc/neutron/fwaas_driver.ini
[fwaas]
driver = neutron_fwaas.services.firewall.drivers.hillstone.hillstone_fwaas.HillstoneFwaas
Driver
enabled = True

[hillstone_fwaas]
hs_manager_addr = 192.168.20.20
hs_manager_http_port = 80
hs_manager_ssh_port = 22
#FWaaS plugin driver NBI model: 1-RestAPI, 2-CLI
fwaas_api_work_model = 1
#FW type: 1-CloudHive, 2-CloudEdge, 3-HW appliance
fwaas_with_vefa_vfw = 3
#whether to support RESTPROXY mechanism
support_sdn_mechanism = False
```

5.4 Limitations

Limitation 1:

This release of Hillstone FWaaS Plugin Driver only supports firewall appliances with StoneOS release 5.5R1 and above.

Limitation 2:

This release of Hillstone FWaaS Plugin Driver only supports firewall configuration for one tenant, as specified during installation.

5.5 Testing

5.5.1 Test cases

The Hillstone firewall appliance is deployed at the OpenStack data center perimeter. The firewall can provide security for North-South traffic between VMs in the data center and

machines on the Internet. Firewall policies configured using the OpenStack FWaaS extension are automatically synchronized to the Hillstone firewall appliance. The following test cases demonstrate how security can be configured on the Hillstone firewall appliance to protect VMs running in the data center.

Test case 1 - Provide security to server VM

This test case demonstrates how firewall policy can be configured to provide security to a server VM. With the right policy configuration, only allowed traffic can access a server VM in the data center. All other traffic is blocked.

This test case includes the following steps:

Step 1: Start a server VM, server1, on the server network. Assign the floating IP, 10.1.1.112 to this server1 VM.

Step 2: Access this floating IP from external network with the following applications: ping, http, and ssh.

Step 3: Configure three firewall rules for server1, add them to a policy, and create a firewall through the FWaaS plugin. The order of these rules will be:

- Allow http to the floating IP

- Allow ping to the floating IP

- Deny all protocols to the floating IP

These rules will only allow http and ping to reach the server1 VM, but block all other protocols.

Step 4: Login to the Hillstone firewall appliance and check if the above FWaaS firewall rules are configured on Hillstone firewall.

Step 5: Access this floating IP from the external network via ping, http, and ssh, and check which protocols are allowed or denied.

Test case 2 – Provide security to host VM

This test case demonstrates how the Hillstone Firewall can provide traffic control for northbound traffic. User VMs in the data center can access the Internet. If certain types of access creates risk for these VMs, this type of access can be blocked by firewall rules.

This test case includes the following steps:

Step 1: Start a user VM, vm1, on the VM network. To reach the Internet, all guest VMs use the IP of router interface, 10.1.1.111. Ping and connect an external ftp server from vm1.

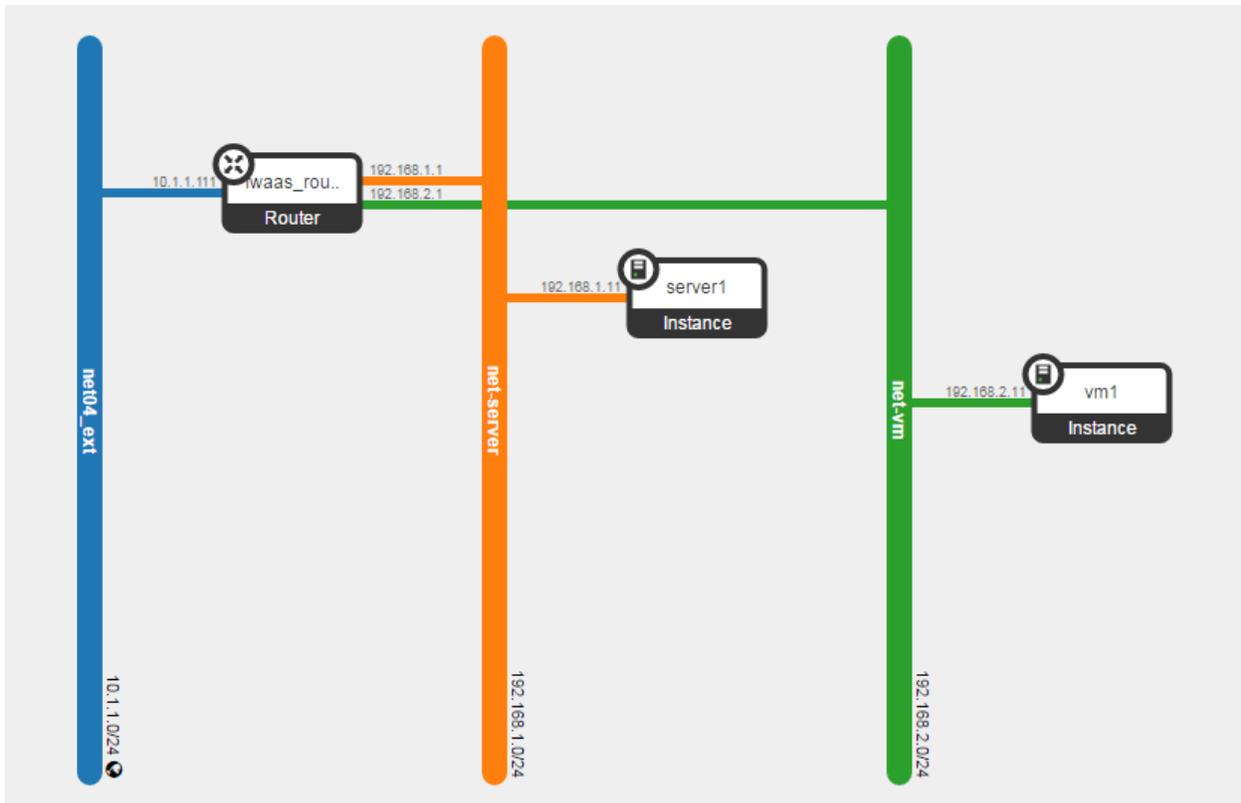
Step 2: Create a firewall rule that blocks the ftp TCP port. Add this rule to the policy.

Step 3: Run the same traffic test as in Step 1.

5.5.2 Test Results

Test case 1 results

Step1: The VM and network configuration is as follows:



Step 2: From a PC on the external network, try to access the floating IP of server1. The traffic result is as follows:

```

50.0.17.74 - PuTTY
$ ping -c 2 10.1.1.112
PING 10.1.1.112 (10.1.1.112) 56(84) bytes of data.
64 bytes from 10.1.1.112: icmp_seq=1 ttl=63 time=1.74 ms
64 bytes from 10.1.1.112: icmp_seq=2 ttl=63 time=1.51 ms

--- 10.1.1.112 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.518/1.630/1.743/0.119 ms
$ wget 10.1.1.112
--2016-04-26 23:02:47-- http://10.1.1.112/
Connecting to 10.1.1.112:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11510 (11K) [text/html]
Saving to: 欽模ndex.html.1欽?

100%[=====>] 11,510  --.-K/s  in 0s

2016-04-26 23:02:47 (150 MB/s) - 欽模ndex.html.1欽? saved [11510/11510]

$ ssh 10.1.1.112
test@10.1.1.112's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-83-generic x86_64)

```

The host on the external network can access the server1 VM through the Hillstone firewall with ping, http, and ssh. At this stage, no firewall policy has been configured.

Step 3: The configuration of FWaaS firewall rules, policy and firewall are as follows:

Firewalls Firewall Policies **Firewall Rules**

+ Add Rule Delete Rules

<input type="checkbox"/>	Name	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action	Enabled	In Policy	Actions
<input type="checkbox"/>	south-server1-http-allow	TCP	-	-	10.1.1.112/32	80	ALLOW	Yes	policy-fwaas	Edit Rule
<input type="checkbox"/>	both-any-any-allow	ANY	-	-	-	-	ALLOW	Yes	policy-fwaas	Edit Rule
<input type="checkbox"/>	south-server1-ping-allow	ICMP	-	-	10.1.1.112/32	-	ALLOW	Yes	policy-fwaas	Edit Rule
<input type="checkbox"/>	south-server1-all-deny	ANY	-	-	10.1.1.112/32	-	DENY	Yes	policy-fwaas	Edit Rule

Displaying 4 items

Firewalls **Firewall Policies** Firewall Rules

+ Add Policy Delete Policies

<input type="checkbox"/>	Name	Rules	Audited	Actions
<input type="checkbox"/>	policy-fwaas	south-server1-http-allow, south-server1-ping-allow, south-server1-all-deny, both-any-any-allow	No	Edit Policy

Displaying 1 item

Firewalls **Firewall Policies** Firewall Rules

+ Create Firewall Delete Firewalls

<input type="checkbox"/>	Name	Policy	Associated Routers	Status	Admin State	Actions
<input type="checkbox"/>	fw-fwaas	policy-fwaas	fwaas_router1	Active	UP	Edit Firewall

Displaying 1 item

Step 4: Login to the Hillstone firewall web management UI. Review the current policy rules:

ID	Name	Sta...	Validity	Source Zo...	Source Address	User/Use...	Destination...	Destination Address	Service	Application
27	south-server1-http-allow	🟢	yes	Any	Any		Any	10.1.1.112/32 (IP add...	tcp-80-80-1-65535	
28	south-server1-ping-allow	🟢	yes	Any	Any		Any	10.1.1.112/32 (IP add...	icmp-1-65535-1-65535	
29	south-server1-all-deny	🟢	yes	Any	Any		Any	10.1.1.112/32 (IP add...	Any	
30	both-any-any-allow	🟢	yes	Any	Any		Any	Any	Any	

This policy configuration page shows that the firewall rules configured via the FWaaS plugin have been automatically transferred to the Hillstone firewall.

Step 5: Run the same traffic test as in Step 2 from the host on the external network. Traffic results are as follows:

```

50.0.17.74 - PuTTY
$ ping -c 2 10.1.1.112
PING 10.1.1.112 (10.1.1.112) 56(84) bytes of data.
64 bytes from 10.1.1.112: icmp_seq=1 ttl=63 time=1.98 ms
64 bytes from 10.1.1.112: icmp_seq=2 ttl=63 time=1.31 ms

--- 10.1.1.112 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.313/1.648/1.983/0.335 ms
$ wget 10.1.1.112
--2016-04-26 23:25:21-- http://10.1.1.112/
Connecting to 10.1.1.112:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11510 (11K) [text/html]
Saving to: 欽模ndex.html.2欽?

100%[=====>] 11,510 --.-K/s in 0s

2016-04-26 23:25:21 (130 MB/s) - 欽模ndex.html.2欽? saved [11510/11510]

$ ssh 10.1.1.112
^C
$

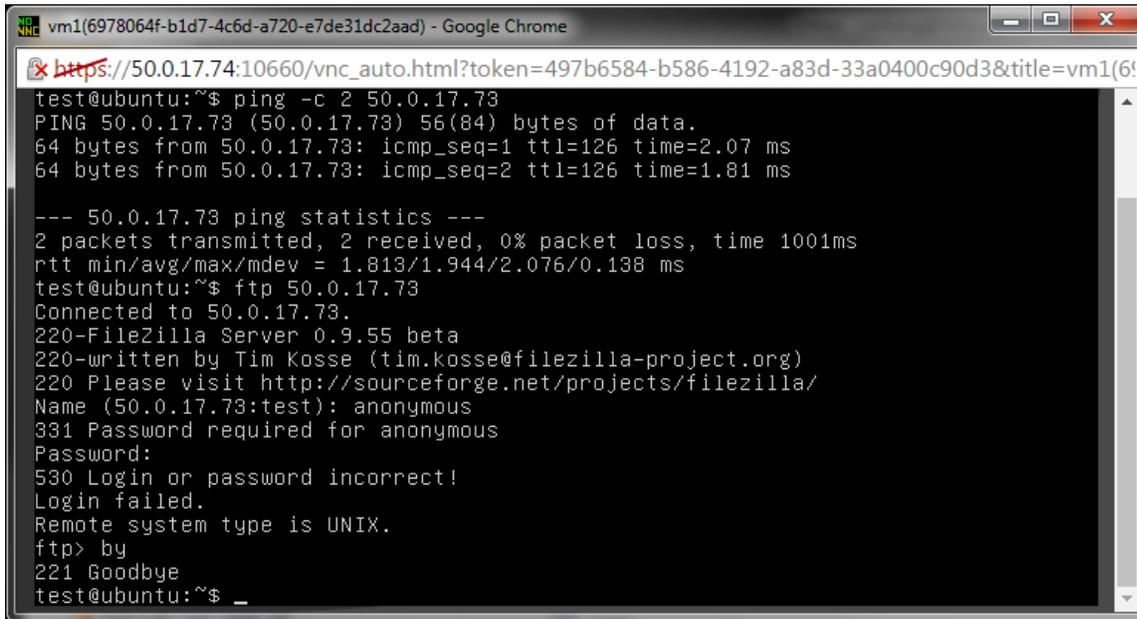
```

Ping and http can now go through the Hillstone firewall since these protocols are allowed by the firewall policy. But ssh cannot connect because it is blocked by a firewall rule.

The above test results demonstrate that firewall rules configured via the OpenStack FWaaS extension can be automatically transferred to the Hillstone firewall appliance by the driver. Data center admins can define firewall configurations using the FWaaS extension and these configurations are implemented on the Hillstone perimeter firewall automatically.

Test case 2 results

Step 1: VM vm1 can ping or ftp to an ftp server on the Internet.



```
vm1(6978064f-b1d7-4c6d-a720-e7de31dc2aad) - Google Chrome
https://50.0.17.74:10660/vnc_auto.html?token=497b6584-b586-4192-a83d-33a0400c90d3&title=vm1(6978064f-b1d7-4c6d-a720-e7de31dc2aad)
test@ubuntu:~$ ping -c 2 50.0.17.73
PING 50.0.17.73 (50.0.17.73) 56(84) bytes of data:
64 bytes from 50.0.17.73: icmp_seq=1 ttl=126 time=2.07 ms
64 bytes from 50.0.17.73: icmp_seq=2 ttl=126 time=1.81 ms

--- 50.0.17.73 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.813/1.944/2.076/0.138 ms
test@ubuntu:~$ ftp 50.0.17.73
Connected to 50.0.17.73.
220-FileZilla Server 0.9.55 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (50.0.17.73:test): anonymous
331 Password required for anonymous
Password:
530 Login or password incorrect!
Login failed.
Remote system type is UNIX.
ftp> by
221 Goodbye
test@ubuntu:~$ _
```

Step 2: A Deny firewall rule is created with 10.1.1.111/32 as the source IP, and TCP port 21 as the destination port. This rule is added into the policy, thus added to the existing firewall.

Firewalls Firewall Policies Firewall Rules

[+ Add Rule](#) [x Delete Rules](#)

<input type="checkbox"/>	Name	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action	Enabled	In Policy	Actions
<input type="checkbox"/>	south-server1-http-allow	TCP	-	-	10.1.1.112/32	80	ALLOW	Yes	policy-fwaas	Edit Rule ▾
<input type="checkbox"/>	both-any-any-allow	ANY	-	-	-	-	ALLOW	Yes	policy-fwaas	Edit Rule ▾
<input type="checkbox"/>	south-server1-ping-allow	ICMP	-	-	10.1.1.112/32	-	ALLOW	Yes	policy-fwaas	Edit Rule ▾
<input type="checkbox"/>	south-server1-all-deny	ANY	-	-	10.1.1.112/32	-	DENY	Yes	policy-fwaas	Edit Rule ▾
<input type="checkbox"/>	north-vm1-ftp-deny	TCP	10.1.1.111/32	-	-	21	DENY	Yes		Edit Rule ▾

Displaying 5 items

Firewalls Firewall Policies Firewall Rules

+ Add Policy Delete Policies

<input type="checkbox"/>	Name	Rules	Audited	Actions
<input type="checkbox"/>	policy-fwaas	north-vm1-ftp-deny, south-server1-http-allow, south-server1-ping-allow, south-server1-all-deny, both-any-any-allow	No	Edit Policy

Displaying 1 item

Step 3: Run the same ping and ftp test again. Ping can go through, but ftp now fails to connect to the remote server, because the control connection is blocked by the Hillstone firewall.

```

vm1(6978064f-b1d7-4c6d-a720-e7de31dc2aad) - Google Chrome
https://50.0.17.74:10660/vnc_auto.html?token=497b6584-b586-4192-a83d-33a0400c90d3&title=vm1(6978064f-b1d7-4c6d-a720-e7de31dc2aad)
test@ubuntu:~$ ping -c 2 50.0.17.73
PING 50.0.17.73 (50.0.17.73) 56(84) bytes of data.
64 bytes from 50.0.17.73: icmp_seq=1 ttl=126 time=2.22 ms
64 bytes from 50.0.17.73: icmp_seq=2 ttl=126 time=1.80 ms

--- 50.0.17.73 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.807/2.017/2.227/0.210 ms
test@ubuntu:~$ ftp 50.0.17.73
^Ctest@ubuntu:~$

```

Test case 2 demonstrates that firewall rules can be created to limit Internet access by data center VMs.